International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 7, Issue 2, Apr 2017, 29-36

ASSOCIATION RULE MINING USING MODIFIED BPSO

AMIT KUMAR CHANDANAN, KAVITA & M K SHUKLA

Department of Computer Science Jayoti Vidyapeeth Women's University, Jaipur, India

ABSTRACT

In this paper, for formulating rules, we are using modified BPSO based on mutation function. Association rules are generated without specifying minimum support and confidence which improves the drawback of apriori, Fp-growth algorithm. In proposed method, the problem of convergence in BPSO is improved using mutation part of Genetic Algorithm and also comparing the results of BPSO and modified BPSO.

KEYWORDS: Frequent Pattern Mining, BPSO & GA

Received: Feb 02, 2017; **Accepted**: Mar 24, 2017; **Published**: Mar 29, 2017; **Paper Id.**: IJCSEITRAPR20175

INTRODUCTION

Association Rule Mining [1, 2] is a technique in Data Mining that is used to reveal the hidden correlation among the different items of transactions exhibit in the database. An association rule can be described as any rule that involves association relationship among different objects (or itemset) such as an object implies to another or the events of these articles alone or with other objects. Association rules [1, 2] are, in general, if-then rules that work on some conditional probability. The two main parameters used for such conditions are support and confidence. The support can be concocted of as the percentage that all the items in the rules will satisfy. The confidence then again can be characterized as the degree of certainty that an association Let in a database D there are a number of transactions T. In each transaction there is number of items having a place with itemset I. If n is the distinct number of items in D then I = {i1, i2…in} is a set of all the items present in database. Also any transaction t ∈ T may contain variable set of items over I, i.e., ii, ij, ik ⊂ I. Each transaction is associated with an interesting identifier. T_ID. The association rule is of the shape of X⇒Y, where X, Y ⊂ I and X Y = ∅, where X is the consequent of the rule.

The association X⇒Y holds for any transaction T in D if its bolster S of any item is satisfied. Support s of an association rule R is the percentage of transaction t that contains XUY (both X and Y) which is the probability P(XUY) of the items in transaction.

Support (X⇒Y) = sup(R) = P(XUY).

The association rule R, of the form X⇒Y has confidence C in transaction set T in D if the contingent likelihood fulfills, i.e., the transaction t containing X also contains Y. It is taken as P(Y/X).

Confidence(c) = confidence (X⇒Y) = conf(R) = P(Y/X) = support_count(XUY)/support_count(X) = sup(R)/sup(X).

An example of an association rule is as follows,

Original Article

Cheese -> beer [sup = 10%, conf = 80%]

This rule says that 10% of clients purchase cheese and beer together, and those who buy cheese additionally purchase beer 80% of the time [3],[4].

Frequent Pattern Mining

Approximate FPM which objectives to find out thrilling and generalized know-how in noisy databases, has received increasingly more pursuits in latest years. The approximate frequent patterns have shown their extreme advantages in various domains, such as discovering approximate association rules[5], reconstructing noisy databases [6-8], and clustering/classification[9,10].

Data tend to be diverse and dirty, all things considered, applications, which may be caused by different factors such as noise, imprecise measurements, network latency and sampling errors. When mining interesting knowledge from these applications, traditional frequent pattern mining(FPM) approaches, such as Apriori[11], FP-growth[12] and Eclat[13] algorithms, are confronted with huge difficulties.

Challenges in Approximate FPM First and for most, the effective FPM algorithms are developed on the basis of anti-monotonicity property, which is feasible for the algorithms to prune applicant patterns and slender the search space. However, the anti-monotonicity property is not accessible in the vast majority of the uproarious available in by far most of the uproarious environments. Therefore, the approximate FPM algorithms have to resort to heuristics-based methods to prune search space, which provides no guarantee on the completeness of the search and only imprecise mining results are obtained.

Moreover, with violation of anti-monotonicity property, the mining of approximate frequent patterns poses new challenges in itemsets' generation. In traditional cases, all nonempty subsets of a regular itemset are also frequent, which is the fundamental to any depth-first approach and leads to the success of FPM algorithms. However, in the field of approximate frequent pattern mining, this property does not hold on candidate generation, which means one cannot obtain the support set of an AFP directly from its sub-patterns. Therefore, multiple scans on the original database are expected to register the support of every itemset. Thus the time unpredictability of the relevant breadth-first algorithms is exponential in the maximum quantity of potential itemsets. Furthermore, the support computation of a candidate itemset has ended up being NP-hard even in the case that only fixed number of error is tolerated in each item [14].

In addition, with missing items in databases, large frequent patterns are broken into short fragments with low support counts, so that the original "true" frequent patterns can't be recuperated by the traditional FPM algorithms. Thus only the fragments are obtained, which are less interesting or informative than the original "true" ones. Consequently, AFP mining algorithms are approved for with the aim of recovering the original embedded true patterns, but fall into a new dilemma. That is, unlike the traditional FPM algorithms to achieve exact frequent patterns, AFP mining approaches are inclined to get the approximatively correct results with false positive errors or false negative errors. If it isn't always handled with caution, unreliable or even incorrect mining results could be obtained [15].

Sequential Pattern Mining

PM proposed by way of Agrawal[16] on analyzing big data from supermarket, is an vital branch of data mining. SPM, crucial branch of data mining. Consecutive example mining, popular in web get to pattern analysis, market basket

analysis, fault detection in network, DNA sequences etc, which needs to find All the sequential pattern that surpasses the base support threshold [17]. Conventional algorithm on sequential pattern mining are classified categories: successive example that outperforms the base support threshold[17]. Traditional calculation on SPM are characterized classes: Apriori, GSP, projection and SPADE [18].

Apriori use codes generating-testing methods and is simple and easy to implement. However, Apriori generates a massive amount of items-sets and scans the database frequently, for that reason wastes a massive amount of time GSP [19] in view of the frequency-item mining algorithm of Apriori and uses time limitations, sliding window to improve the efficiency while it needs traverse the database multiple times.SPADE[20] by Zaki transforms the data into a vertical form,but generates masses of items-sets. Generating item-sets and branch trimming consumes extraordinary measure of time. Based on projection, Freespan[21] and Prefix span[22] use “divide-conquer” to divide the raw database into smaller projection databases, and then mine the sequential pattern in smaller databases. Divide conquer builds the productivity and has excellent expansion. However, this method spends incredible measure of time in dividing database into projection databases and has the bottle neck in constructing projection databases and scanning data [23].

Hierarchical Database

Another sort of database is emerging both in the research community and in the commercial market place. This new sort of database permits designers to speak to hierarchical data in XML form while providing query, transaction and security services similar to commercial relational database software. (In fact, hierarchical databases are not new; some earlier databases used hierarchical data models such as CODASYL. The hierarchical version is taking part in a rebirth with the arrival of XML [24]). While it will most likely not replace all relational databases, it is being aimed at just the sort of problem we have defined in implementing the NMP missions and technology database. The database canonically implements the data hierarchy. A hierarchy, in this case, can be thought of as a tree structure. An example of such a structure that is natural to the majority of the community structure that is recognizable to the greater some portion of the group is the file system directory, as seen in the Windows or Macintosh stack of folders metaphor. Here, parent or higher-level folders may contain child, or lower-level folders, which, in tum, contain folders of yet a lower level. Every organizer may likewise contain a specific type of data or file. If used as intended (hut not enforced), "child folders", i.e., those contained within their "parent folder" contain data that is a subset of the types of data contained in the parent folder. In addition, pointers (aliases or "shortcuts") are provided to allow linking logically connected, but non-adjacent, folders. In most cases, the folders are displayed, not as a tree, but as an indentured list. While the indentured list is a convenient and space-efficient format, it does, unfortunately, tend to bide the tree structure. Still it is a familiar construct with which most individuals are familiar, and for which the underlying structure is readily grasped.

The upsides of a hierarchical database are basically the inverse of the drawbacks of the relational database, i.e.With a hierarchical database, the hierarchy is the native structure. There is no need to craft custom interfaces to hide the actual database structure or to interpret it for the user. Hierarchical data is stored in a hierarchical format (XML). A simple display interface permits the client guide access to the structure as implemented and the data as stored. System maintenance and debug efforts are much reduced. In this business, surprises are not good, and this ability to view the structures as they are tends to minimize surprises [25].

Literature Survey

Table 1: Littérature Review

S. No	Author Name	Algorithm	Proposed Work
1.	Ruilin Liu [2016] et. al [26]	SLAM algorithm	An efficient rare association rule mining algorithm called spark-based rare association rule mining (SRAM) which leverages not only the efficiency of FP-growth algorithm but also the powerful big data processing mechanism of spark platform. We have implemented our algorithm on the start platform and tested with various of data sets.
2.	Morteza Zihayat [2016] et. al [27]	*BigHUSP*	Another structure for mining HUSPs in tremendous data. A dispensed and parallel algorithm referred to as Big HUSP is proposed to discover HUSPs efficiently. At its heart, Big HUSP makes use of multiple Map Reduce-like steps to process information in parallel. We also propose some of pruning techniques to reduce seek area in disbursed surroundings, and consequently decrease computational and communique charges, whilst nevertheless preserving correctness.
3.	Mohammad Karim Sohrabi [2016] et. al [28]	CUSE algorithm	a novel bit astute way to deal with pack and speak to the sequence database as a 3-dimentional array and use a corresponding mining method to extract frequent sequences from the compressed structure Experimental results and overall performance observe display that this calculation beats the best formerly evolved algorithms.
4.	Nicolle Chaves Cysneiros [2016] et. al [29]		A solution in which an ontology's reasoner is used to retrieve the subclasses of the authentic queried concepts. These retrieved concepts are used to rewrite the submitted query
5.	Aneesh K. Sahu [2015] et. al [29]	Cryptography algorithm	The proposed replica capably to find global frequent item sets even when no site can be treated as trusted. The trusted party initiates the process and prepares thmerged list.
6.	Hong-Yi Chang [2015] et. al [31]	Apriori algorithm and FP-Growth algorithm	We developed a method that combines the Apriori and FP-Growth algorithms with MapReduce to rectify this problem. In experiments carried out, we varied the block length of the Mapper to obtain execution performance higher than the ones of the Apriori and FP-Growth algorithms
7.	Masome sadat Hoseini [2015] et. al [32]	FP-growth algorithm	A new approach is presented for mining Cantree, and it's evaluated to reveal its development over the FP-growth technique that mine FP tree.

Problem Statement

The existing technique generates association rules by binary particle swarm optimization,which has the low convergence problem. Due to this problem the execution time of algorithm increase and also the association rules generated are more which requires more memory for storage.

Proposed Work and Result Analysis

In the proposed methodology, binary particle swarm optimization is used with mutation function, through which the low convergence problem of BPSO can become less. It can be done by applying a suitable mutation rate. The combined

approach of BPSO with mutation function generates rule with less execution time and has no convergence problem.

Table 2: Base Results (Elapsed Time Is 2.078145 Seconds)

Rule Number Antecedent	Consequent	Support Confidence
Herring Heineken, Corned-B	85.00	85.00
Herring Soda	75.00	75.00
Soda Heineken, Avacado	65.00	86.67
P. Frames Avacado	45.00	90.00
Soda,ArtichokeHeineken, Baugette	45.00	90.00
Baugette, Olives Avacado	93.33	93.33
Cracker, Baugette P. Frames, Turkey	15.00	33.33
Cracker, Heineken, Avacado, BaugetteSoda, Herring	35.00	100.00

Above table shows the association rules of the form: Antecedent-> consequent(Support %, Confidence %)

These rules are generated by applying binary particle swarm optimization. The Elapsed time of base algorithm when executed on MATLAB is 2.078145 seconds.

Table 3: Results With Proposed Experiment (Elapsed Time Is 3.634799 Seconds)

Rule Number Antecedent	Consequent	Support Confidence
Avacado Cracker	45.00	56.25
Soda Avacado,Baugette	70.00	93.33
AvacadoCracker,Turkey	45.00	56.25
Soda,P. Frames Herring	45.00	100.00

Above table shows the association rules of the form: Antecedent-> consequent(Support %, Confidence %)

These rules are generated by applying binary particle swarm optimization with mutation operator. The Elapsed time of proposed algorithm when executed on MATLAB is 1.246828 seconds.

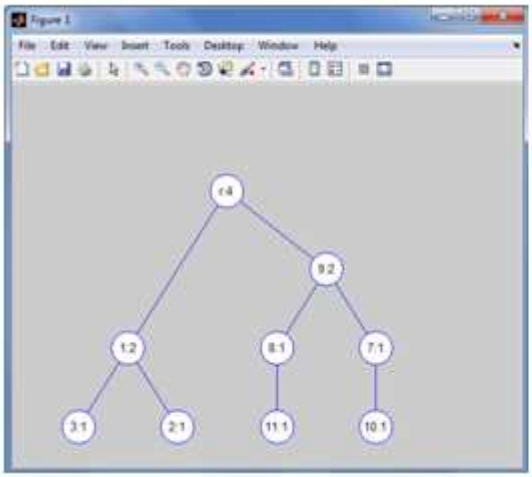

Figure 1: Simulation

Figure 2: Simulation Result

The above figure shows outcome of program execution.

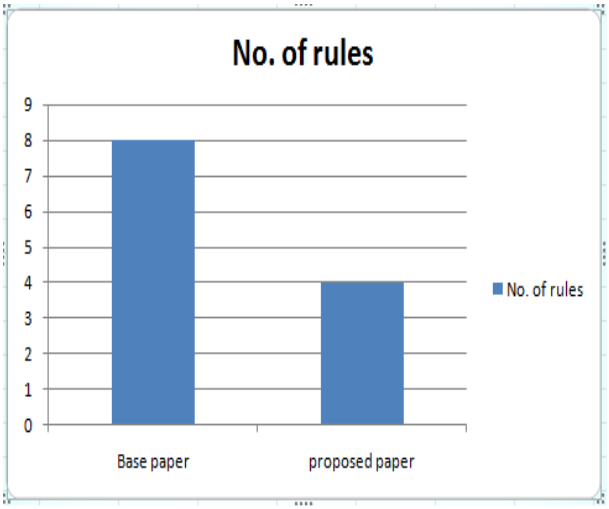

Figure 1: Rule Comparison

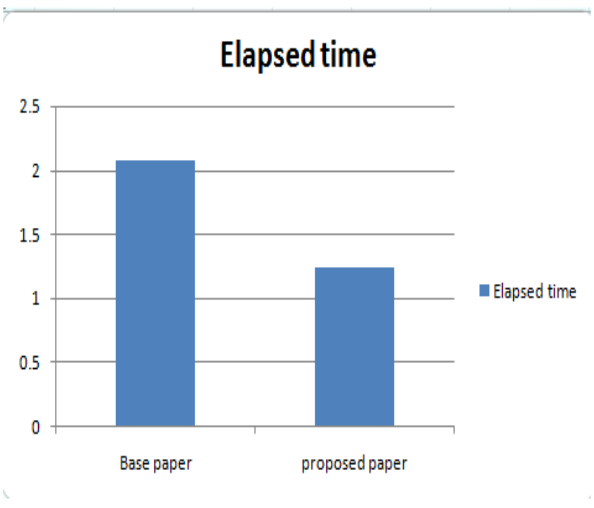

Figure 2: Elapsed Time Comparison

The above rule comparison shows association rules that are generated by base algorithm are more and rules generated by proposed approach are less.

The above elapsed time comparison is showing that our proposed approach is executed in less time as compared to the base values.

CONCLUSIONS

In this paper, rules are formed by using BPSO merged with mutation, which doesnot generate redundant rules and also improves the convergence problem of BPSO. The comparison of elapsed time and number of rules is also given. Our proposed methodology generates less number of rules with no redundancy and less number of elapsed time in comparison with other algorithm.

ACKNOWLEDGEMENTS

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Kavita,Professor, Computer Science, Jayoti Vidyapeeth Women's University,Jaipur and my research co-supervisor Dr. M K Shukla,Professor, Computer Science, Jayoti Vidyapeeth Women's University,Jaipur for giving me the opportunity to do research and providing invaluable guidance throughout this research. their dynamism, vision, sincerity and motivation have deeply inspired me. She has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy, and great sense of humor. I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my wife and my daughters for their love, understanding, prayers and continuing support to complete this research work.

My Special thanks goes to my present working istitute that supports me enhance my career. Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

REFERENCES

1. *Agrawal, R., Imielinski, T., Swami, A. "Mining association rules between sets of items in large databases." pp. 207-216, ACM SIGMOD 1993*
2. *Agrawal, R. and Srikant, R. "Fast algorithms for mining association rules." Proceedings of VLDB, 1994.*
3. *Feng Tao, Fionn Murtagh & Mohsen Farid, "Weighted Association Rule Mining using Weighted Support and Significance Framework", ACM SIGKDD 2003.*
4. *Satpal Singh, Vivek Badhe, "Profit Association Rule Mining with Inventory Measures", 978-1-5090-0076-0/15 $31.00 © 2015 IEEE*
5. *J. Pei, A. K. H. Tung, and J. Han. Fault-tolerant frequent pattern mining: Problems and challenges. In Workshop on Research Issues in DMKD, 2001.*
6. *J. Liu, S. Paulsen, X. Sun, W. Wang, A. B. Nobel, and J. Prins. Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In SDM 2006, pp.405-416.*
7. *J. K. Seppanen and H. Mannila. Dense itemsets. In KDD'04, 2004, pp. 683-688.*
8. *J. Besson, R. G. Pensa, C. Robardet, and J.-F. Boulicaut. Constraintbased mining of fault-tolerant patterns from boolean data. In KDID, 2005, pp.55-71.*
9. *H. Cheng, P. S. Yu, and J. Han. Approximate frequent itemset mining in the presence of random noise. In Soft Computing for Knowledge Discovery and Data Mining, 2007, pp. 363-389.*
10. *R. Gupta, G. Fang, B. Field, M. Steinbach, and V. Kumar. Quantitative evaluation of approximate frequent pattern mining algorithms. In KDD, 2008, pp.301-309.*
11. *R. Agrawal and R. Srikant. Fast algorithms for mining association rules. Proceedings 20th International Conference on Very Large Data Bases(VLDB'94), September 12-15, 1994, pp.487̧499.*
12. *J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM Press, 2000, pp. 1-12.*
13. *M.J.Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, Vol.3, 2000, pp.372-390.*
14. *A.K.Poernomo, Gopalkrishnan, V.:Mining statistical information of frequent faulttolerant patterns in transactional databases. In: ICDM 2007, 2007, pp. 272-281.*
15. *Xiaomei Yu, Yongqin Li, Hong Wang, "Mining Approximate Frequent Patterns From Noisy Databases", 978-1-4673-8315-8 /15 $31.00 © 2015 IEEE*
16. *R. Agrawal, R. Srikant. Mining Sequential Pattern[C]//Pro. of the 11st Int. Conf. on Data Engineering, Taipei,1995,3:3~14*
17. *HAN J. KAMBER M. Concept and Technology of Data Mining [M] . Fan Ming,Meng Xiao-feng Translation . BeiJing : Machinery Industry Press. 2001 : 320—336.*
18. *Chen Zhuo,Yang Rui-ru,Song Wei,Song Ze-feng. Survey of sequential pattern mining[J]. APPLICATION RESEARCH OF COMPUTERS.,2008,07:1960-1963+1976.*

19. *Srikant R,Agrawal R. Mining sequential pattems: generali-zations and performance improvements [C]. EDBT 96:Proceeding of the 5th International Conference on Extending Database Technology: Advance in Database Technology .UK,London :Springer-Verlag,1996:3-17*
20. *Zaki M. SPADE: An Efficient Algorithm for Mining Frequent Sequence [J] .Macheine Learning,2001,42(1):31-60.*
21. *Han J, Pei J, Mortazavi-Asl B, et al. Freespan : freguent patternprojected seguential pattern minin〔A〕.In : Proceedings of the International Conference on Knowledge Discovery and Data Mining ACM〔C〕.Montreal, Canada, 2000. 355 359.*
22. *Pei J,Han J,Mortazavi-Asl B,et al. Mining sequential patterns by patterngrowth: the prefixspan approach [J]. IEEE Transaction On Knowledge and Data Engineering,2004,16(11):1424-1440.*
23. *XUE Fei and SHAN Zheng, "A Improved Sequential Pattern Mining Algorithm Based on PrefixSpan", xue2016*
24. *S. Hong, "Introduction to Database Management Systems,"http://www.cis.gsu.edu/-shong/cis8 14/sli des/intro1 .pdf.*
25. *Raphael R. Some, Akos Czikmantory, "XML Hierarchical Database for Missions and Technologies", 0-7803-81 55-6/04/$17.00 02004 IEEE*
26. *Ruilin Liu, Kai Yang, Yanjia sun, Tao Quan, Jin Yang, " spark based Rare Association Rule Mining for big Datasets", 978-1-4673-9005-7/16/$31.00 ©2016 IEEE 2734.*
27. *Morteza Zihayat, Zane Zhenhua Hu, Aijun An, Yonggang Hu, "Distributed and Parallel High Utility Sequential Pattern Mining", 978-1-4673-9005-7/16/$31.00 ©2016 IEEE.*
28. *Mohammad Karim Sohrabi and Vahid Ghods, "CUSE: A Novel Cube-based Approach for Sequential Pattern Mining", 978-1-5090-3488-8/16/$31.00 ©2016 IEEE*
29. *Nicolle Chaves Cysneiros, Ana Carolina Salgado, "Including Hierarchical Navigation in a Graph Database Query Language with an OBDA Approach", 978-1-5090-2109-3/16/$31.00 © 2016 IEEE.*
30. *Aneesh K. Sahu, Raghvendra Kumar, Naazish Rahim, "Mining Negative Association Rules in Distributed Environment", 978-1-5090-0076-0/15 $31.00 © 2015 IEEE.*
31. *Hong-Yi Chang, Yih-Jou Tzang, Jia-Chi Lin, Zih-Huan Hong, Ting-Yun Chi, Chun-Yen Huang, "A Hybrid Algorithm for Frequent Pattern Mining Using MapReduce Framework", 978-1-4673-8600-5/15 $31.00 © 2015 IEEE.*
32. *Masome sadat Hoseini, Mohammad Nadimi Shahraki, Behzad Soleimani Neysiani, "A new algorithm for mining frequent patterns in CanTree",978-1-*41-4673-6506-2/15/2015IEEE.